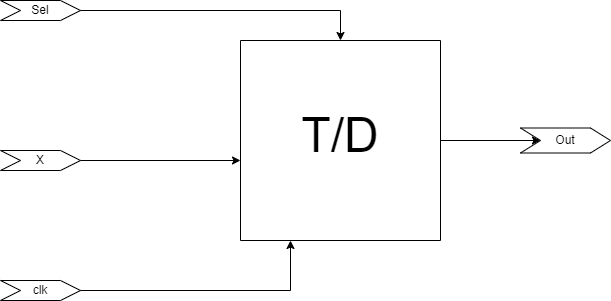
# Progetto di Un Flip Flop Custom che possa commutare da T a D in base ad un segnale di selezione, sincronizzato sul fronte di salita del clock e suo successivo utilizzo in una macchina sincrona

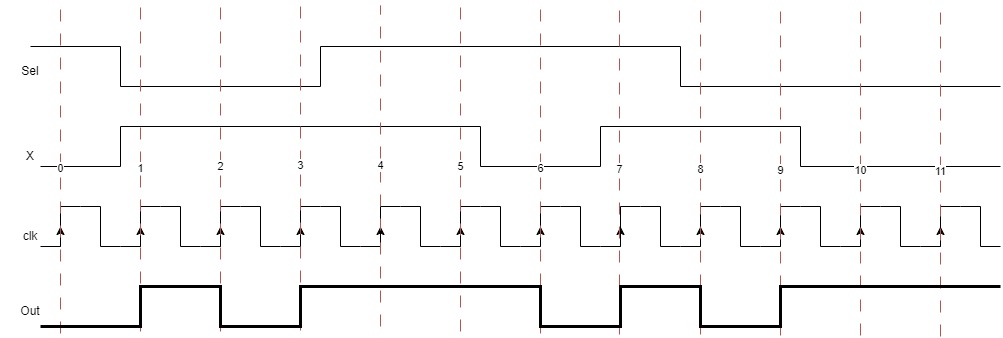
Come da titolo si voglia inizialmente progettare un FF “custom”. Tale dispositivo possa commutare il suo funzionamento da FFT a FFD in base ad un segnale di selezione. Oltre tale segnale il FF sarà dotato anche di un segnale di clock sul cui fronte di salita si verrà a sincronizzare e da un segnale di ingresso “X” che controllerà le transizioni in modo coerente al funzionamento selezionato.

* Se il segnale di Sel è a livello **basso** il dispositivo si comporti come un **FF T**
* Se il segnale di Sel è a livello **alto** il dispositivo si comporti come un **FF D**

Individuare i segnali coinvolti:

Visto dall’esterno il sistema si presenta dotato di 3 ingressi e di un’uscita:  
- un segnale a livelli *Sel* che definisce la modalità di funzionamento (come detto sopra)  
- un segnale *clk* i cui fronti di salita scandiscono gli istanti nei quali il FF viene a commutare  
- un segnale a livelli X che fornisce l’ingresso di controllo del FF e opera in base a quanto selezionato su Sel.

.

Descrizione sintetica   
Comprendere pienamente il funzionamento del dispositivo, meglio se attraverso un esempio che evidenzi un ipotetico segnale di stimoli (composto dai segnali Sel, clk, X) e la risposta attesa. Tali stimoli potranno poi eventualmente essere impiegati in fase di simulazione per verificare se il funzionamento ottenuto è conforme a quello previsto.

**D**

**D**

**T**

**T**

1. All’istante 0 il segnale Sel=1 indica un funzionamento da FF-D, il quale vede in ingresso un segnale X a livello basso e pertanto pone l’uscita ad un valore identico.
2. All’istante 1 Sel=0 indica un funzionamento da FF-T e con X=1 si ha una commutazione dell’uscita
3. Gli istanti 2 e 3 similmente all’istante 1 portano successive commutazioni dell’uscita
4. Negli istanti 4 e 5 il dispositivo funziona come un FF-D e quindi campiona il segnale d’ingresso e lo riporta in uscita che quindi sarà alta
5. All’istante 6 il dispositivo funziona come un FF-D, il segnale X è però basso e così sarà pure l’uscita
6. All’istante 7 il segnale Sel indica un funzionamento da FF-D e la presenza di un segnale X=1 all’ingresso porta il medesimo segnale in uscita
7. All’istante 8 e 9 il dispositivo torna a funzionare come FF-T e pertanto con segnale di ingresso pari a 1 e pertanto ad ogni fronte di clock l’uscita viene a commutare
8. Negli istanti 10 e 11 Sel=0 (FF-T) ma X=0 per cui l’uscita rimane inalterata.

Evidenziare i meccanismi di funzionamento

Vi sono due modi nei quali si può procedere:

**Metodo 1**: Individuare quali sequenze di ingressi comportano una certa variazione dell’uscita:

Partendo dal fatto che nella macchina che si andrà a realizzare vi saranno due stati di cui uno relativo all’uscita **bassa** ed uno che mantiene l’uscita **alta**, si può indagare quali siano le sequenze di segnali sugli ingressi atte a far sì che la macchina raggiunga questi stati:

1. Perché l’uscita passi da livello 0 al livello 1 è necessario che:
   1. O il dispositivo funzioni da FFT (Sel=0), sia presente un valore alto sull’ingresso X e arrivi un fronte di salita sul clock:

ovvero CSX : 001 🡪 101

* 1. O il dispositivo funzioni da FFD (Sel=1), sia presente un valore alto sull’ingresso X e arrivi un fronte di salita sul clock:

ovvero CSX : 011 🡪 111

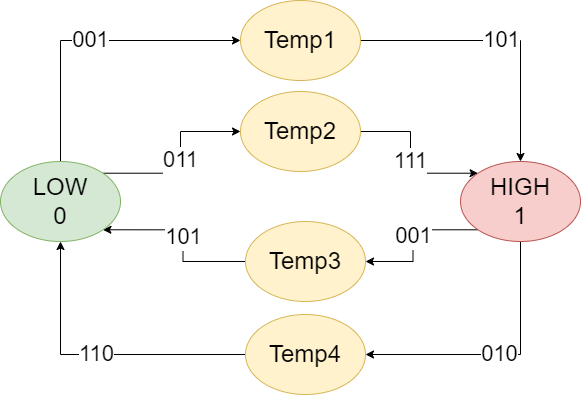
1. Perché l’uscita passi da livello 1 al livello 0 è necessario che:
   1. O il dispositivo funzioni da FFT (Sel=0), sia presente un valore alto sull’ingresso X e arrivi un fronte di salita sul clock:

ovvero CSX : 001 🡪 101

* 1. O il dispositivo funzioni da FFD (Sel=1), sia presente un valore basso sull’ingresso X e arrivi un fronte di salita sul clock:

ovvero CSX : 010 🡪 110

Questo funzionamento può essere schematizzato nella seguente macchina di Moore che evidenzia solo le transizioni complete che portano la macchina a passare dallo stato LOW allo stato HIGH passando attraverso degli opportuni stati intermedi, che mantengono l’uscita inalterata fintanto che la transizione non sia completata.



Passando quindi ad una rappresentazione secondo la Tavola di Huffman essa potrebbe inizialmente essere abbozzata come segue:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| LOW | LOW | TMP1 | TMP2 | LOW | LOW | LOW | LOW | LOW | 0 |
| TMP1 |  | TMP1 |  |  |  | HIGH |  |  | 0 |
| TMP2 |  |  | TMP2 |  |  |  | HIGH |  | 0 |
| HIGH | HIGH | TMP3 | HIGH | TMP4 | HIGH | HIGH | HIGH | HIGH | 1 |
| TMP3 |  | TMP3 |  |  |  | LOW |  |  | 1 |
| TMP4 |  |  |  | TMP4 |  |  |  | LOW | 1 |

E successivamente completata tenendo conto che laddove la transizione non dovesse completarsi perché i segnali di eccitazione fossero cambiati prima dell’arrivo del fronte di clock e le condizioni non fossero più coerenti con la commutazione dell’uscita la macchina dovrebbe rimettersi nello stato originario oppure in un diverso stato temporaneo di attesa. Le condizioni NON raggiungibili (in quanto si presuppone che i segnali di ingresso vengano ad essere modificati uno alla volta) sono lasciate nella condizione ‘-‘ (don’t care)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| LOW | LOW | TMP1 | TMP2 | LOW | LOW | LOW | LOW | LOW | 0 |
| TMP1 | LOW | TMP1 | TMP2 | - | - | HIGH | - | - | 0 |
| TMP2 | - | TMP1 | TMP2 | LOW | - | - | HIGH | - | 0 |
| HIGH | HIGH | TMP3 | HIGH | TMP4 | HIGH | HIGH | HIGH | HIGH | 1 |
| TMP3 | HIGH | TMP3 | HIGH | - | - | LOW | - | - | 1 |
| TMP4 | HIGH |  | HIGH | TMP4 | - | - | - | LOW | 1 |

Ad un’attenta analisi della macchina appena realizzata si può però notare che gli stati TMP1 e TMP2 siano compatibili e quindi semplificabili ed altrettanto dicasi degli stati TMP3 e TMP4, portando quindi alla versione semplificata:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| LOW | LOW | TMP12 | TMP12 | LOW | LOW | LOW | LOW | LOW | 0 |
| TMP12 | LOW | TMP12 | TMP12 | - | - | HIGH | HIGH | - | 0 |
| HIGH | HIGH | TMP34 | HIGH | TMP34 | HIGH | HIGH | HIGH | HIGH | 1 |
| TMP34 | HIGH | TMP34 | HIGH | TMP34 | - | LOW | LOW | - | 1 |

**Metodo 2**: Realizzare una macchina che ad ogni variazione del clock cambi SEMPRE di stato, raggiungendo uno stato la cui uscita sia coerente con le variabili di eccitazione presenti in ingresso al momento in cui è arrivato il fronte di clock.

Quando si può individuare un **unico** segnale di sincronismo tale per cui la macchina reagisca solamente quando su questo segnale avviene una qualsivoglia variazione del suo stato (fronte positivo, fronte negativo, impulso, oppure la presenza tanto di fronte positivo quanto di un fronte negativo (DDR)), allora si può pensare di realizzare un dispositivo che mantenga il suo stato, indipendentemente dal valore assunto dagli altri ingressi e che modifichi opportunamente detto stato **solo quando** viene rilevata una certa variazione del segnale di clock. Nel caso in esame si può pensare di realizzare una macchina dotata quindi di 4 stati di cui due con uscita **alta** (C e D) e due con uscita **bassa** (A e B) ed al contempo due stabili per quando il clock è a livello **basso**(A e C) ed i restanti due, stabili quando il clock è a livello **alto** (B e D).

Uno “scheletro” di questa macchina è il seguente:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| A | A | A | A | A |  |  |  |  | 0 |
| B |  |  |  |  | B | B | B | B | 0 |
| C | C | C | C | C |  |  |  |  | 1 |
| D |  |  |  |  | D | D | D | D | 1 |

È evidente che fintanto non si rilevino variazioni sul clock la macchina rimarrà stabile in uno dei quattro stati indipendentemente da eventuali variazioni sui segnali S ed X. Per completare le caselle in bianco bisogna però adesso ragionare sul fatto che quando arriva un fronte positivo di clock l’evoluzione passerà da uno stato stabile posto nella parte sinistra della tabella verso uno stato stabile posto a destra, e viceversa quando si dovesse rilevare un fronte negativo del clock. Sarà pertanto cura del progettista far evolvere la macchina verso lo stato che presenta l’uscita idonea in base agli altri segnali di eccitazione come da specifiche del progetto.

Ad esempio supponiamo che il clock sia allo stato basso e la macchina sia nello stato A. Qui vi rimane indipendentemente da come vengono modificati i segnali S e X e pertanto la macchina NON cambia stato. Supponiamo i valori di eccitazione in ingresso si siano ora assestati al valore SX=01 quando arriva un fronte positivo sul clock. I valori idi eccitazione (S e X) indicano che la macchina deve comportarsi come farebbe un FF-T con l’eccitazione posta a 1, ovvero deve invertire l’uscita, la transizione corretta sarà pertanto quella che fa evolvere la macchina verso lo stato D, che presenta appunto l’uscita alta e dove vi rimarrà fintanto che il clock non si sarà nuovamente abbassato.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| A | A | A | A | A |  | D |  |  | 0 |
| B |  |  |  |  | B | B | B | B | 0 |
| C | C | C | C | C |  |  |  |  | 1 |
| D |  |  |  |  | D | D | D | D | 1 |
|  |  |  |  |  |  |  |  |  |  |

Ora poichè si è scelto di realizzare un dispositivo che fosse sensibile solo ai fronti di salita del clock, i fronti di discesa NON devono alterare l’uscita indipendentemente da quali siano gli altri segnali di eccitazione ecco pertanto che all’arrivo del fronte di discesa del clock la macchina dovrà evolvere **sempre** da B verso A oppure da D verso C che sono gli stati che mantengono l’uscita inalterata.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| A | A | A | A | A |  | D |  |  | 0 |
| B | A | A | A | A | B | B | B | B | 0 |
| C | C | C | C | C |  |  |  |  | 1 |
| D | C | C | C | C | D | D | D | D | 1 |

Con analoghi ragionamenti si possono completare quindi tutte le altre transizioni (relative ai fronti positivi del clock) scegliendo via via se si vuole attivare l’uscita passando allo stato D oppure disattivarla passando allo stato B.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| A | A | A | A | A | B | D | D | B | 0 |
| B | A | A | A | A | B | B | B | B | 0 |
| C | C | C | C | C | D | B | D | B | 1 |
| D | C | C | C | C | D | D | D | D | 1 |

Scelta di un’opportuna codifica degli Stati e realizzazione della logica

A questo punto per realizzare le macchine progettate si deve definire una codifica degli stati che possibilmente eviti l’insorgenza di corse critiche garantendo che durante le transizioni si venga a modificare una sola variabile di stato alla volta.

Per la **Macchina 1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| LOW | LOW | TMP12 | TMP12 | LOW | LOW | LOW | LOW | LOW | 0 |
| TMP12 | LOW | TMP12 | TMP12 | - | - | HIGH | HIGH | - | 0 |
| HIGH | HIGH | TMP34 | HIGH | TMP34 | HIGH | HIGH | HIGH | HIGH | 1 |
| TMP34 | HIGH | TMP34 | HIGH | TMP34 | - | LOW | LOW | - | 1 |

si può scegliere ad esempio LOW: 00 – TMP12:01 – HIGH: 11 – TMP34:10

il che porta a definire la seguente tavola di flusso

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| y1y2\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| 00 | 00 | 01 | 01 | 00 | 00 | 00 | 00 | 00 | 0 |
| 01 | 00 | 01 | 01 | - | - | 11 | 11 | - | 0 |
| 11 | 11 | 10 | 11 | 10 | 11 | 11 | 11 | 11 | 1 |
| 10 | 11 | 10 | 11 | 10 | - | 00 | 00 | - | 1 |

E quindi la logica per calcolare l’uscita, nonché le due variabili di stato

Ovvero per l’uscita:

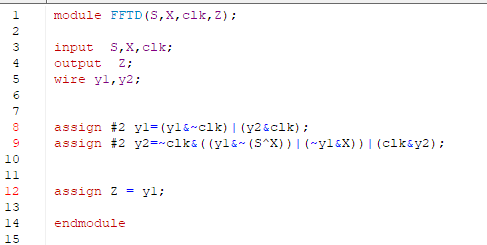
Per ***Y1***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| 00 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | - | - | 1 | 1 | - |
| 11 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 10 | 1 | 1 | 1 | 1 | - | 0 | 0 | - |

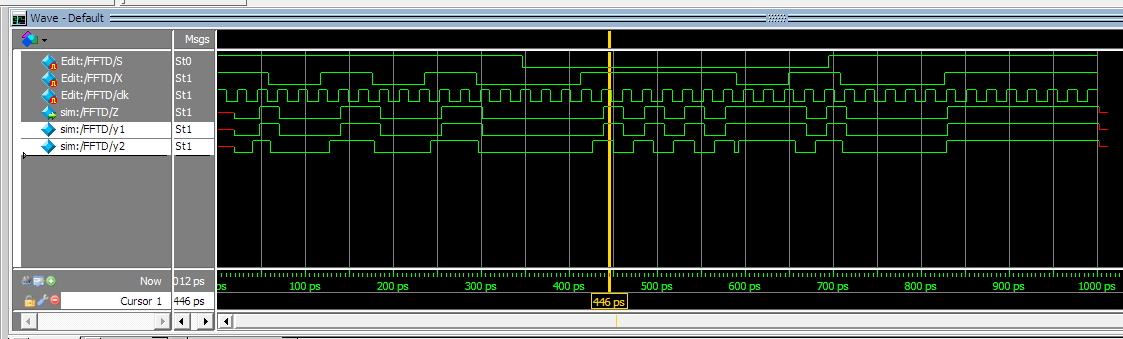
Per ***Y2***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| 00 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 01 | 0 | 1 | 1 | - | - | 1 | 1 | - |
| 11 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 | - | 0 | 0 | - |

Che possono essere opportunamente verificate attraverso idonea simulazione



Onde verificare il corretto funzionamento



Per la **Macchina 2**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| A | A | A | A | A | B | D | D | B | 0 |
| B | A | A | A | A | B | B | B | B | 0 |
| C | C | C | C | C | D | B | D | B | 1 |
| D | C | C | C | C | D | D | D | D | 1 |

si può scegliere ad esempio A: 00 – B:01 – C: 11 – D:10

il che porta a definire la seguente tavola di flusso:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 | OUT |
| 00 | 00 | 00 | 00 | 00 | 01 | 10 | 10 | 01 | 0 |
| 01 | 00 | 00 | 00 | 00 | 01 | 01 | 01 | 01 | 0 |
| 11 | 11 | 11 | 11 | 11 | 10 | 01 | 10 | 01 | 1 |
| 10 | 11 | 11 | 11 | 11 | 10 | 10 | 10 | 10 | 1 |

E quindi la logica per calcolare l’uscita, nonché le due variabili di stato

Ovvero per l’uscita:

Per ***Y1***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| 00 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| 01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|  |  |  |  |  |  |  |  |  |

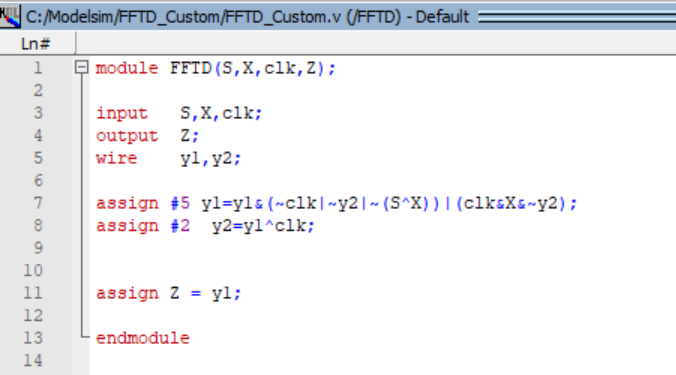
ovvero

Mentre per ***Y2***

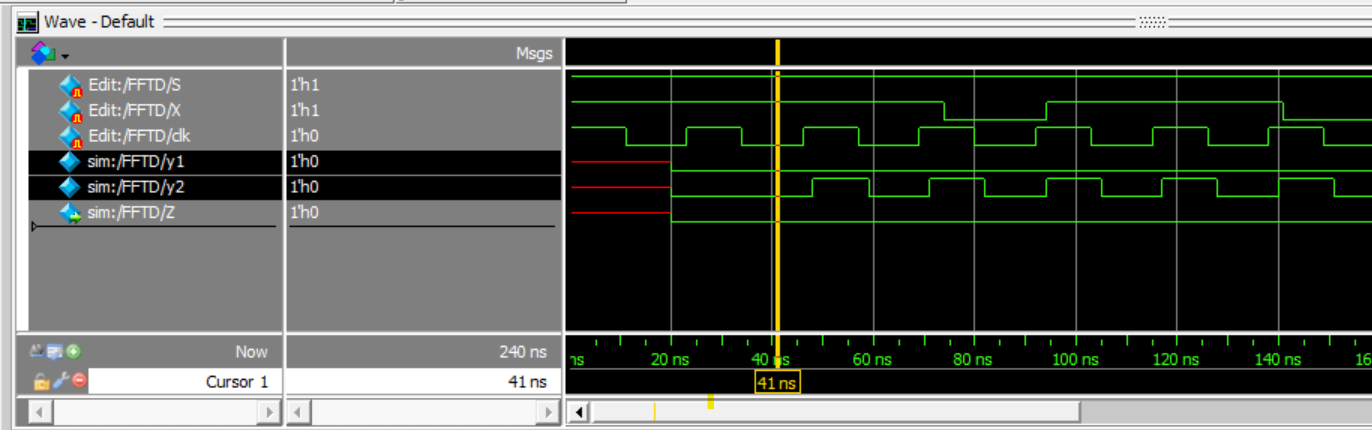
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| St\CSX | 000 | 001 | 011 | 010 | 100 | 101 | 111 | 110 |
| 00 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 01 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 11 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 10 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Si può notare che la mappa di Karnaugh di Y2 per la metà di sinistra coincide con quella di Y1 mentre per la metà di destra è l’opposto (la negazione) di Y1, da cui si può desumere che

Bisogna però prestare attenzione al fatto che sebbene le equazioni trovate siano fondamentalmente giuste una loro realizzazione attraverso delle porte logiche che rappresentino pedissequamente le operazioni individuate potrebbero creare dei problemi al circuito finale: Questa seconda equazione richiede infatti che y2 venga calcolata sulla base di y1, ovvero **dopo** che questo sia stato calcolato, ma l’applicazione di una singola porta XOR posta tra il clock e la variabile y1 rischia di presentare un tempo di ritardo inferiore a quello della logica che deve valutare y1. Se così dovesse accadere y2 si modificherebbe al giungere del fronte di clock in base al valore precedente di y1 e questo si propagherebbe verso la logica stessa che calcola y1 prima che essa si sia aggiornata, rischiando di fatto un’alea. Un esempio di questo malfunzionamento può essere evidenziato attraverso la simulazione:

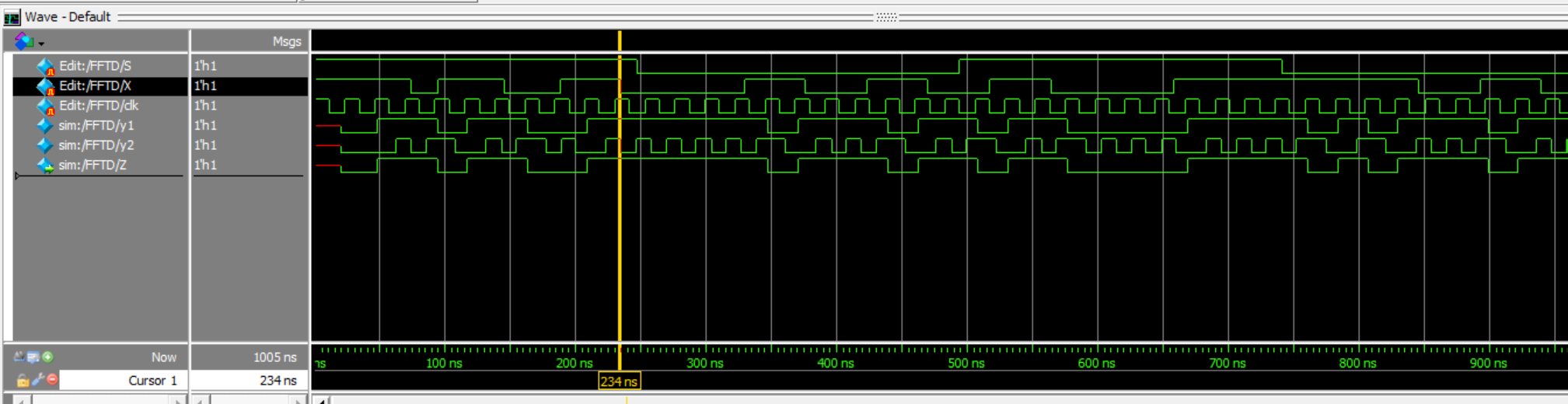


Supponiamo infatti che il ritardo con cui viene elaborato y2 sia inferiore a quello necessario per y1.



La simulazione inizia correttamente con un valore sconosciuto. All’istante 20ns le variabili di stato vengono manualmente forzate ad un valore noto (y1y2=00) e successivamente svincolate all’istante 40ps. Ora la macchina si trova nello stato A:00 con ingressi CSX=011. Quando arriva il fronte positivo del clock (all’istante 46) secondo la tavola di flusso lo stato dovrebbe passare al valore 10=D (ovvero dovrebbe commutare y1 e y2 dovrebbe rimanere inalterata), ma poiché il tempo di ritardo per calcolare y2 è inferiore a quello per calcolare y1, quella che viene a modificarsi per prima è y2 e lo fa sulla base del valore già presente su y1. Y2 pertanto viene portata al valore 1 e da qui si propaga immediatamente agli ingressi della logica che calcola y1 (descritta con l’assegnazione al rigo 7 del codice VerilogHDL). La logica deputata al calcolo di y1 (e che non ha ancora completato il calcolo) vede ora in ingresso i valori Y1=0 e Y2=1 e attraverso questi stabilisce che il valore sul quale deve assestarsi sia il valore 0 (come da tavola di flusso). Ed ecco pertanto che la macchina invece di evolvere verso lo stato 10=D evolve verso lo stato 01=B creando di fatto un malfunzionamento.

Se invece invertissimo ad esempio i valori dei ritardi tra le due funzioni logiche garantendo che Y1 commuti prima di Y2 e che quest’ultima si assesti sul valore attuale di Y1 ecco che la macchina funzionerebbe correttamente.

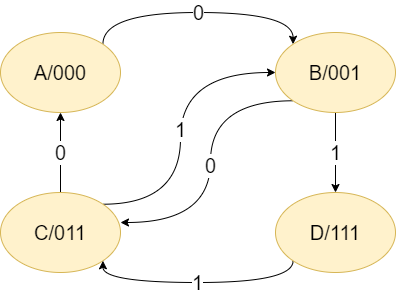


Va ovviamente sottolineato che i ritardi di propagazione non possono essere controllati a piacimento e dipendono caso per caso. Questo esercizio è volto a sottolineare come una corretta progettazione non si debba fermare alle equazioni da realizzare, ma anche le scelte circuitali rivestono un ruolo fondamentale nel buon funzionamento del dispositivo.

# Impiego del FF in una macchina sincrona.

Utilizziamo ora il FF appena progettato per realizzare le celle di memoria all’interno di una macchina sincrona, cercando di sfruttare al meglio le sue caratteristiche.

Supponiamo si voglia realizzare la seguente macchina di Moore composta da un ingresso e tre uscite:



Si inizi col descriverla attraverso una tavola di Huffman

|  |  |  |  |
| --- | --- | --- | --- |
| St\Ing | 0 | 1 | Out |
| A | B | - | 000 |
| B | C | D | 001 |
| C | A | B | 011 |
| D | - | C | 111 |

Si Introduca ora una codifica per gli stati. In questo caso essendovi 4 Stati distinti si possono impiegare due variabili di stato, ad esempio A:00, B:01, C:10, D:11 arrivando quindi a definre la tavola di flusso.

|  |  |  |  |
| --- | --- | --- | --- |
| y1y2\Ing | 0 | 1 | z1,z2,z3 |
| 00 | 01 | - | 000 |
| 01 | 10 | 11 | 001 |
| 10 | 00 | 01 | 011 |
| 11 | - | 10 | 111 |

Attraverso essa si può innanzitutto definire le equazioni che definiscono le tre uscite desiderate in base alle variabili di stato.

Adesso si passi ad analizzare le transizioni delle due variabili di stato:

|  |  |  |
| --- | --- | --- |
| st|Ing | 0 | 1 |
| 00 | 01 | - |
| 01 | 10 | 11 |
| 10 | 00 | 01 |
| 11 | - | 10 |

Dove si sono indicati col colore nero quei livelli (0 o 1) che rimangono inalterati durante la transizione ed in rosso quelli che viceversa hanno cambiato di livello.

Per ottenere queste transizioni nelle memorie che memorizzano lo stato della macchina sincrona, si deve agire sui segnali di eccitazione dei FF, ma vi sono diverse possibilità per ottenere ciascuna transizione:

Per meglio comprendere quali siano le eccitazioni che inducono una particolare transizione torniamo ad analizzare il comportamento del FF Custom e vediamo come i sui segnali di eccitazione inducono le varie transizioni, attraverso una descrizione del FF attraverso la tavola di flusso che ne descrive il funzionamento.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| st\SX | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |

Dalla precedente tavola di flusso si desume che per garantire che lo stato del FF

* Rimanga a 0 (0 nero) il segnale X deve essere a livello basso ma S può assumere qualsiasi livello
* Passi da 1 a 0 (0 rosso) il segnale X può essere sia 0 che 1 purchè il segnale S sia l’opposto di X
* Rimanga a 1 (1 nero) il segnale X può essere sia 0 che 1 purchè il segnale S sia uguale a X
* Passi da 0 a 1 (1 rosso) il segnale S può essere sia 0 che 1 purchè il segnale X sia posto a 1

Questi funzionamenti possono essere sintetizzati nella seguente tabella di eccitazione:

|  |  |  |
| --- | --- | --- |
| Trans\eccit | S | X |
| 0 | - | 0 |
| 0 | 0/1 | 1/0 |
| 1 | 0/1 | 0/1 |
| 1 | - | 1 |

Evidenziando ora le transizioni compiute da ciascuna variabile di stato durante l’evoluzione della macchina si può calcolare come debbano essere poste le variabili di eccitazione per indurre tale transizione

Per il primo FF

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tr-FF1 |  |  |  | S1 |  |  |  | X1 |  |  |
| y1y2|Ing | 0 | 1 |  | y1y2|Ing | 0 | 1 |  | y1y2|Ing | 0 | 1 |
| 00 | 0 | - |  | 00 | - | - |  | 00 | 0 | - |
| 01 | 1 | 1 |  | 01 | - | - |  | 01 | 1 | 1 |
| 10 | 0 | 0 |  | 10 | 0/1 | 0/1 |  | 10 | 1/0 | 1/0 |
| 11 | - | 1 |  | 11 | - | 0/1 |  | 11 | - | 0/1 |

Mentre per il secondo FF

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tr-FF2 |  |  |  | S2 |  |  |  | X2 |  |  |
| y1y2|Ing | 0 | 1 |  | y1y2|Ing | 0 | 1 |  | y1y2|Ing | 0 | 1 |
| 00 | 1 | - |  | 00 | - | - |  | 00 | 1 | - |
| 01 | 0 | 1 |  | 01 | 0/1 | 0/1 |  | 01 | 1/0 | 0/1 |
| 10 | 0 | 1 |  | 10 | - | - |  | 10 | 0 | 1 |
| 11 | - | 0 |  | 11 | - | 0/1 |  | 11 | - | 1/0 |

Le tabelle sopra esposte indicano che si possono fare diverse scelte nella realizzazione delle funzioni responsabili di generare le eccitazioni per i due FF, ma bisogna fare particolare attenzione a non confondere la situazione evidenziata con “1/0” oppure “0/1” con la condizione “-“ ossia “don’t care”. In entrambi i casi infatti il progettista può scegliere a piacere se l’uscita da realizzare con quei particolari ingressi sia da porre a 0 oppure a 1, ma mentre nel caso “-“ questa scelta è totalmente libera, negli altri casi essa è vincolata a fare una scelta omologa anche nella tabella relativa alla seconda variabile di eccitazione. Per fare un esempio pratico, nelle tabelle relative ad S2 e X2, relativamente agli ingressi y1,y2,x=1,1,1 si può scegliere se S2 debba assumere il valore 0 (ma allora saremmo costretti a scegliere per X2, con i medesimi ingressi il valore 1) oppure possiamo scegliere se porre l’uscita 1 (ma allora saremmo costretti a scegliere per X2, sotto i medesimi ingressi, il valore 0).

Viceversa se guardiamo al caso y1,y2,x=0,0,0 l’uscita della logica che calcola X2 dovrà obbligatoriamente essere posta ad 1, ma siamo assolutamente liberi di scegliere per la logica che calcola S2 l’uscita più idonea, ovvero che semplifichi al meglio la realizzazione finale.

Una possibile scelta per la definizione della logica che pilota le variabili di eccitazione dei due Flip Flop potrebbe essere ad esempio quella evidenziata in blu nelle seguenti tabelle. Si noti ancora una volta che si può segliere caso per caso sia l’uscita di “destra” che quella di “sinistra” ma ove sceglie l’uscita “di destra relativamente ad una variabile di eccitazione, altrettanto andrà fatto anche per la seconda variabile di eccitazione ed ovviamente lo stesso dicasi se si sceglie l’uscita di “sinistra”

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S1 |  |  |  | X1 |  |  |
| y1y2|Ing | 0 | 1 |  | y1y2|Ing | 0 | 1 |
| 00 | - | - |  | 00 | 0 | - |
| 01 | - | - |  | 01 | 1 | 1 |
| 10 | 0/1 | 0/1 |  | 10 | 1/0 | 1/0 |
| 11 | - | 0/1 |  | 11 | - | 0/1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S2 |  |  |  | X2 |  |  |
| y1y2|Ing | 0 | 1 |  | y1y2|Ing | 0 | 1 |
| 00 | - | - |  | 00 | 1 | - |
| 01 | 0/1 | 0/1 |  | 01 | 1/0 | 0/1 |
| 10 | - | - |  | 10 | 0 | 1 |
| 11 | - | 0/1 |  | 11 | - | 1/0 |

Da cui si possono ottenere le seguenti funzioni, particolarmente semplici, per le variabili di eccitazione:

Prima di passare alla simulazione del dispositivo così progettato è bene indagare su quale sarà il suo funzionamento semmai esso dovesse finire nello stato A con ingresso pari a 1 oppure nello stato D con ingresso pari a 0. Queste condizioni nelle specifiche di progetto erano di tipo “*don’t care”*, ma per comprendere al meglio il completo funzionamento del dispositivo è bene indagare su quale sarebbe l’evoluzione semmai in fase di utilizzo il sistema si ponesse erroneamente in tali condizioni.   
In alternativa bisogna essere particolarmente cauti acciocché tali condizioni siano evitate.

Dalle equazioni trovate per le eccitazioni si nota che

* Se il sistema fosse nello stato A (y1y2=00) con Ingresso pari a 1

S1 varrebbe 0 ed X1 varrebbe 0, quindi la prima variabile di stato Y1 non commuterebbe

S2 varrebbe 1 ed X2 varrebbe 1, quindi Y2 verrebbe a commutare,, quindi lo stato futuro sarebbe 01 ovvero B

* Se il sistema fosse nello stato D (y1y2=11) con Ingresso pari a 0

S1 varrebbe 1 ed X1 varrebbe 1, quindi la prima variabile di stato Y1 non commuterebbe

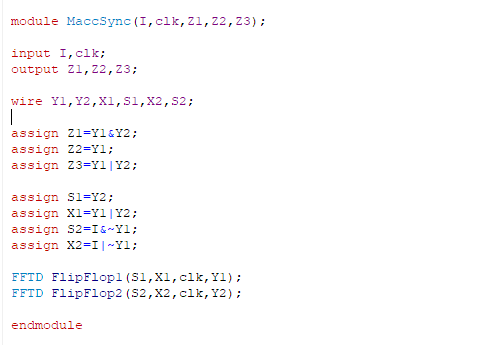
S2 varrebbe 0 ed X2 varrebbe 0, quindi nemmeno Y2 verrebbe a commutare, ovvero quindi lo stato futuro sarebbe 11 ovvero D stesso

Il funzionamento completo della macchina sarebbe pertanto riassunto nella seguente tavola di flusso

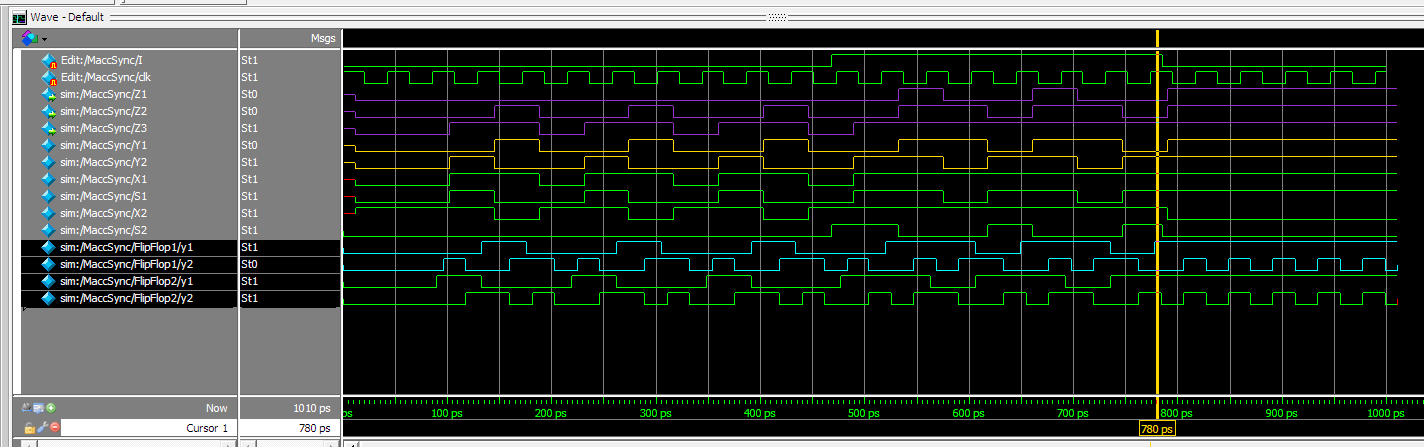
|  |  |  |  |
| --- | --- | --- | --- |
| y1y2\Ing | 0 | 1 | z1,z2,z3 |
| 00 | 01 | 01 | 000 |
| 01 | 10 | 11 | 001 |
| 10 | 00 | 01 | 011 |
| 11 | 11 | 10 | 111 |

Questo ci sta ad indicare che semmai in fase di simulazione si decidesse di portare l’ingresso al valore 0 quando la macchina si trovasse nello stato D, essa qui vi si fermerebbe indefinitamente fintanto che l’ingresso non fosse riportato al valore 1.

Si può quindi passare a descrivere il circuito in VerilogHDL instanziando opportunamente i moduli del FF realizzati nella parte iniziale dell’esercizio.



Ed eseguirne la simulazione.



Dove si vede anche un esempio del comportamento citato poco fa, nel quale la macchina, in particolari condizioni, si stabilizza nello stato D, condizione peraltro non prevista nelle specifiche di progetto.

Una particolare attenzione fa fatta nella “forzatura” delle condizioni iniziali. Non avendo introdotto alcun meccanismo di reset la macchina, appena accesa potrebbe trovarsi in qualsiasi condizione (evidenziata con StX) e da questa evolverebbe sempre verso altrettante condizioni sconosiute. Si deve pertanto forzare una condizione iniziale. Questa va fatta “manualmente” ponendo il dispositivo in un ben definito stato di partenza, però questa “forzatura” non va limitata agli stati della macchina sincrona, anzi la forzatura in uno di tali stati risulta abbastanza inutile, ma essa piuttosto va fatta a livello gerarchico inferiore, ovvero intercettando i valori delle variabili di stato dei singoli FF e forzando quest’ultime in uno stato noto per poi sbloccarle e far evolvere naturalmente il dispositivo (si notino le ultime quattro righe del diagramma che servono appunto ad intercettare questi segnali interni per forzarli nella parte iniziale e per vederne l’evoluzione successivamente):